

(19) 日本国特許庁 (JP)

(12) 特 許 公 報 (B2)

(11) 特許番号

特許第5384323号  
(P5384323)

(45) 発行日 平成26年1月8日 (2014.1.8)

(24) 登録日 平成25年10月11日 (2013.10.11)

(51) Int.Cl.

F I

G 0 6 F 9/38 (2006.01)

G 0 6 F 9/38 3 3 O E

G 0 6 F 9/38 3 3 O B

請求項の数 25 (全 15 頁)

(21) 出願番号 特願2009-501686 (P2009-501686)  
 (86) (22) 出願日 平成19年3月19日 (2007.3.19)  
 (65) 公表番号 特表2009-530754 (P2009-530754A)  
 (43) 公表日 平成21年8月27日 (2009.8.27)  
 (86) 国際出願番号 PCT/US2007/064331  
 (87) 国際公開番号 W02007/109631  
 (87) 国際公開日 平成19年9月27日 (2007.9.27)  
 審査請求日 平成20年11月14日 (2008.11.14)  
 (31) 優先権主張番号 11/378, 712  
 (32) 優先日 平成18年3月17日 (2006.3.17)  
 (33) 優先権主張国 米国 (US)

(73) 特許権者 595020643  
 クォアルコム・インコーポレイテッド  
 QUALCOMM INCORPORATED  
 アメリカ合衆国、カリフォルニア州 92  
 121-1714、サン・ディエゴ、モア  
 ハウス・ドライブ 5775  
 (74) 代理人 100108855  
 弁理士 蔵田 昌俊  
 (74) 代理人 100091351  
 弁理士 河野 哲  
 (74) 代理人 100088683  
 弁理士 中村 誠  
 (74) 代理人 100109830  
 弁理士 福原 淑弘

最終頁に続く

(54) 【発明の名称】 複数のビットを有する分岐ヒストリレジスタにおいてループ分岐を表すこと

(57) 【特許請求の範囲】

【請求項 1】

分岐予測の方法であって、

プロセッサが、ループに関連する分岐命令を識別することと、前記識別することは、分岐命令アドレスと実行された最終的分岐命令のアドレスとを比較することを含み、

前記ループに関連する後方分岐命令が識別されたとき、前記プロセッサが、前記後方分岐命令に関連するループの反復の回数を更新し、前記ループの反復が終了するとき、前記プロセッサが、前記ループの反復の回数を示す多ビット値を分岐ヒストリレジスタ (BHR) に記憶することと、

を備える方法。

10

【請求項 2】

前記ループに関連する分岐命令の分岐の方向は後方分岐である、請求項 1 に記載の方法。

【請求項 3】

前記識別することは、前記分岐命令の PC が、最終分岐命令の PC を記憶する最終分岐 PC (LBPC) レジスタの内容とマッチングする場合、前記分岐命令は前記ループに関連する後方分岐命令であると識別する、請求項 1 に記載の方法。

【請求項 4】

前記識別することは、前記分岐命令の PC が、複数の最終分岐命令の PC を記憶する複数の LBPC レジスタのうちのどれかの内容とマッチングする場合、前記分岐命令は前記

20

ループに関連する後方分岐命令であると識別する、請求項 1 に記載の方法。

【請求項 5】

前記分岐命令は、終了ループについてコンパイラによって生成されるユニークな分岐命令である、請求項 1 に記載の方法。

【請求項 6】

前記分岐命令の o p コードは、それが前記ループに関連する後方分岐命令であることを指し示す 1 つまたは複数のビット、を含む、請求項 1 に記載の方法。

【請求項 7】

多ビット値を前記 B H R に記憶することは、あらかじめ決定された数のビットを前記 B H R に記憶することを備える、請求項 1 に記載の方法。

10

【請求項 8】

前記プロセッサが、ループ反復の回数の、前記多ビット値に対する固定されたマッピングに応じて、前記あらかじめ決定された数のビットの前記値を決定すること、をさらに備える請求項 7 に記載の方法。

【請求項 9】

前記プロセッサが、ループ反復の回数の、前記多ビット値に対するスケーラブルなマッピングに応じて、前記あらかじめ決定された数のビットの前記値を決定すること、をさらに備える請求項 7 に記載の方法。

【請求項 10】

前記プロセッサが、ループ反復の回数の、前記多ビット値に対する前記スケーラブルなマッピングを決定するために、複数のしきい値を読み取ること、をさらに備える請求項 9 に記載の方法。

20

【請求項 11】

多ビット値を前記 B H R に記憶することは、前記ループの反復の回数に応じて変化するビットの可変な数を前記 B H R に記憶することを備える、請求項 1 に記載の方法。

【請求項 12】

前記プロセッサが、前記 B H R の下位の複数のビットでループ反復をカウントすること、をさらに備える請求項 11 に記載の方法。

【請求項 13】

前記プロセッサが、ループカウンタにおいてループ反復をカウントすることと、  
前記プロセッサが、前記ループが終了するときに前記ループカウンタの前記値を前記 B H R に転送することと、をさらに備える請求項 11 に記載の方法。

30

【請求項 14】

ループに関連する分岐命令を識別することは、第 1 のループに関連する第 1 の後方分岐命令と、第 2 のループに関連する第 2 の後方分岐命令を検出することを備え、前記第 1 のループは、前記第 2 のループ内でネストされる、請求項 1 に記載の方法。

【請求項 15】

多ビット値を前記 B H R に記憶することは、前記第 1 のループの反復の回数を示す多ビット値を前記 B H R に記憶することを備え、前記第 2 のループの反復の回数を示す多ビット値を前記 B H R に記憶することをさらに備える、請求項 14 に記載の方法。

40

【請求項 16】

多ビット値を前記 B H R に記憶することは、前記第 1 のループの反復の回数を示す多ビット値だけを前記 B H R に記憶することを備える、請求項 14 に記載の方法。

【請求項 17】

多ビット値を前記 B H R に記憶することは、前記第 2 のループの反復の回数を示す多ビット値だけを前記 B H R に記憶することを備える、請求項 14 に記載の方法。

【請求項 18】

条件付き分岐命令の評価を予測するように動作する分岐予測器と、  
前記分岐予測器からの予測に基づいて命令を推論的にフェッチし実行するように動作する命令実行パイプラインと、

50

条件付き分岐命令の評価を記憶するように動作する分岐ヒストリレジスタ ( B H R ) と、  
コードループの反復の回数をカウントするように動作するループカウンタ ( L C ) と、  
条件付き分岐命令に関連するループの反復の回数を示す多ビット値を前記 B H R に記憶  
するように動作する制御回路と、

を備え、

前記制御回路は、ループに関連する条件付き分岐命令を識別し、前記ループに関連する  
条件付き分岐命令が識別されたとき前記 L C を更新し、前記条件付き分岐命令に関連する  
ループの反復の終了のときに、前記多ビット値を前記 B H R に記憶し、前記識別すること  
は、前記条件付き分岐命令のアドレスと実行された最終の条件付き分岐命令のアドレスと  
を比較することを含む、

10

プロセッサ。

【請求項 19】

条件付き分岐命令の P C を記憶するように動作する最終分岐 P C ( L B P C ) レジスタ  
をさらに備え、前記制御回路は、前記分岐命令の P C が前記 L B P C の内容とマッチング  
する場合に、条件付き分岐命令がループに関連づけられることを決定する、請求項 18 に  
記載のプロセッサ。

【請求項 20】

2 つ以上の L B P C レジスタと、対応する 2 つ以上の L C と、をさらに備え、第 1 の L  
B P C は、第 1 のループに関連する条件付き分岐命令の P C を記憶するように動作し、第  
1 の L C は、前記第 1 のループの反復のカウントを保持するように動作し、第 2 の L B P  
C は、第 2 のループに関連する条件付き分岐命令の P C を記憶するように動作し、第 2 の  
L C は、前記第 2 のループの反復のカウントを保持するように動作し、前記第 1 のループ  
は、前記第 2 のループ内でネストされる、請求項 19 に記載のプロセッサ。

20

【請求項 21】

前記制御回路は、前記第 1 のループの 1 つの実行の反復の回数を示す多ビット値と、前  
記第 2 のループの反復の回数を示す多ビット値とを、前記 B H R に記憶するように動作す  
る、請求項 20 に記載のプロセッサ。

【請求項 22】

前記制御回路は、前記第 1 のループの 1 つの実行の反復の回数を示す多ビット値を前記  
B H R に記憶するように、そして前記第 2 のループの反復の回数の表示を前記 B H R に記  
憶しないように動作する、請求項 20 に記載のプロセッサ。

30

【請求項 23】

前記制御回路は、前記第 2 のループの反復の回数を示す多ビット値を前記 B H R に記憶す  
るように、そして前記第 1 のループの任意の実行の反復の回数の表示を前記 B H R に記憶  
しないように動作する、請求項 20 に記載のプロセッサ。

【請求項 24】

前記 B H R は、前記 B H R に直接にループ反復カウントを保持するために、前記ループ  
に関連する前記条件付き分岐命令のおののの分岐成立評価に応じて、複数のビットを増  
分するように動作する、請求項 18 に記載のプロセッサ。

40

【請求項 25】

複数のしきい値に応じて、ループ反復カウントを固定された多ビット値に対してマッピ  
ングするように動作するしきい値化ロジック、をさらに備える請求項 18 に記載のプロセ  
ッサ。

【発明の詳細な説明】

【技術分野】

【0001】

本開示は、一般にプロセッサの分野に関し、特に、複数のビット (multiple bits) を有  
する分岐ヒストリレジスタ (branch history register) においてループ分岐 (loop branche  
s) を表す方法に関する。

50

## 【背景技術】

## 【0002】

マイクロプロセッサは、多種多様なアプリケーションにおいて計算タスクを実行する。改善されたプロセッサ性能は、ソフトウェア変更を介してより高速なオペレーションおよび/または増大された機能を可能にするためにほとんどの場合に望ましい。ポータブル電子デバイスなどの一般的な埋込みアプリケーションにおいては、パワーを節約することも望ましい。

## 【0003】

一般的な現代のプロセッサは、パイプライン化アーキテクチャを使用しており、ここでおののが複数の実行ステップを有するシーケンシャル命令が、実行においてオーバーラップされる。最大の性能のために、命令は、絶えずパイプラインを介して流れるべきである。命令がパイプラインにおいて機能停止する(stall)ようにさせるどのような状況も、性能に悪影響を及ぼす。命令がパイプラインからフラッシュされ、後で再フェッチされる必要がある場合、性能もパワー消費も影響を受けてしまう可能性がある。

## 【0004】

一般にすべての現実の世界のプログラムは、条件付き分岐命令を含み、この条件付き分岐命令の実際の分岐動作(branching behavior)は、命令がパイプラインにおいて深く評価されるまで、一般には知られていない。分岐命令の実際の評価を待つことからもたらされる可能性がある機能停止を回避するために、一般的な現代のプロセッサは、何らかの形態の分岐予測を使用し、それにより条件付き分岐命令の分岐動作は、パイプラインにおいて早期に予測される。予測分岐評価(predicted branch evaluation)に基づいて、プロセッサは、推論的に(speculatively)フェッチし(プリフェッチし(prefetch))、予測されたアドレス(predicted address) - 分岐ターゲットアドレス(分岐が成立と予測される場合(if the branch is predicted taken))かまたは分岐命令の後の次の順次アドレス(分岐が不成立と予測される場合(if the branch is predicted not taken)) - からの命令を実行する。実際の分岐動作が決定されるとき、分岐が誤って予測された場合には、推論的にフェッチされた命令は、パイプラインからフラッシュされる必要があり、新しい命令は正しい次のアドレスからフェッチされる。誤った分岐予測に応じて命令をフェッチすることは、プロセッサ性能およびパワー消費に悪影響を及ぼす。その結果、分岐予測の精度(accuracy)を改善することが望ましい。

## 【0005】

知られている分岐予測技法は、静的予測と動的予測の両方を含んでいる。一部の分岐命令の可能性のある動作は、プログラムおよび/またはコンパイラによって静的に予測されることができる。一例は、エラー検査ルーチン(error checking routine)である。一般的なコードは適切に実行し、エラーは、まれである。したがって、「エラー上分岐(branch on error)」ファンクションをインプリメントする分岐命令は、非常に高いパーセンテージの時間に「不成立(not taken)」と評価することになる。そのような命令は、分岐条件の一般的に可能性のある結果の知識を用いて、プログラムまたはコンパイラによって設定される、opコード(op code)の中の静的分岐予測ビットを含むことができる。

## 【0006】

動的予測は、一般に予測されている分岐命令および/または同じコードにおける他の分岐命令の分岐評価ヒストリ(そして一部の場合には分岐予測精度ヒストリ)に基づいている。実際のコードの広範囲にわたる解析は、最近の過去の分岐評価パターンが、将来の分岐命令の評価のよいインジケータとすることができることを指し示す。

## 【0007】

図1に示される、知られている一形態の動的な分岐予測は、過去のn個の分岐評価を記憶する分岐ヒストリレジスタ(Branch History Register)(BHR)100を利用する。簡単なインプリメンテーションにおいては、BHR30は、シフトレジスタを備える。一般的な最近の分岐評価結果は、レジスタの中の最も以前の過去の評価(the oldest past evaluation)が移動させられて、(例えば、1が利用されることを指し示し、0が利用されな

10

20

30

40

50

いことを指し示して)シフトされる。プロセッサは、各分岐命令についてローカルBHR 100を保持することができる。代わりに(または追加して)、BHR 100は、時に当技術分野においてグローバルBHR、またはGHRとして知られているすべての条件付き分岐命令についての最近の過去の評価を含むことができる。ここにおいて使用されるように、BHRは、ローカル分岐ヒストリレジスタと、グローバル分岐ヒストリレジスタの両方を意味する。

#### 【0008】

図1に示されるように、BHR 100は、分岐予測器テーブル(Branch Predictor Table)(BPT) 102をインデックス付けすることができ、この分岐予測器テーブルは、この場合にもローカルまたはグローバルとすることができる。BHR 100は、BPT 102に直接にインデックス付けすることができ、あるいはBPTインデックスロジック104における分岐命令のプログラムカウンタ(Program Counter)(PC)など、他の情報と組み合わせられることができる。BPTインデックスロジック104に対する他の入力があり、さらに利用されることもできる。BPTインデックスロジック104は、それらの入力を連結し(当技術分野においてgselectとして知られている)、それらの入力のXOR(gshare)を行い、ハッシュ関数(hash function)を実行し、あるいは様々なやり方でそれらの入力を組み合わせ、または変換することができる。

#### 【0009】

一例として、BPT 102は、複数の飽和カウンタ(saturation counters)を備えることができ、これらの飽和カウンタのMSBは、二項の分岐予測器としての役割を果たす。例えば、各テーブルエントリは、おののちに以下の重み付けされた予測値(weighted prediction value)が割り当てられた4つの状態のうちの1つを仮定する(assumes)2ビットカウンタ、を備えることができる。

#### 【0010】

- 11 - 強い予測の成立(Strongly predicted taken)
- 10 - 弱い予測の成立(Weakly predicted taken)
- 01 - 弱い予測の不成立(Weakly predicted not taken)
- 00 - 強い予測の不成立(Strongly predicted not taken)

カウンタは、対応する分岐命令が「成立(taken)」を評価するたびに、増分し(increments)、命令が「不成立(not taken)」を評価するたびに、減分する(decrements)。カウンタのMSBは、二項分岐予測器であり、それは、基礎となる予測の強度または重みにかかわらず、成立の予定である、または不成立の予定である分岐を予測することになる。飽和カウンタは、頻度の低い分岐評価方向の予測エラーを低減させる。一方法を一貫して評価する分岐は、カウンタを飽和させることになる。頻度の低い評価は、他方、カウンタ値(および予測の強度)を変更することになるが、二項予測値を変更することにはならない。それ故に、頻度の低い評価は、1回だけ誤った予測を行うにすぎないことになり、2回は行わない。飽和カウンタのテーブルは、例示の例であるにすぎず、一般に、BHTは、様々な分岐予測メカニズムを含むテーブルにインデックス付けすることができる。

#### 【0011】

BPT 102において使用される分岐予測メカニズムにかかわらず、BHR 100は、単独で、または分岐命令PCなど、他の情報と組み合わせて、分岐予測を取得するためにBPT 102にインデックス付けする。BHR 100に先行した分岐評価を記憶することと、分岐予測における評価を使用することにより、予測されている分岐命令は、過去の分岐動作(ローカルBHR 100の場合におけるそれ自体の過去の動作と、グローバルBHR 100の場合における他の分岐命令の動作)に相関させられる。この相関は、少なくとも高い反復コードの場合には、正確な分岐予測に対する鍵である。

#### 【0012】

図1は、BHR 100に記憶されている分岐評価(-すなわち、実行パイプステージなどにおけるパイプラインの中の深くにおいて知られているだけの可能性がある条件付き分岐命令の実際の評価)を示すことに注意すべきである。これは、最終的な結果であるが、

実際には、一般的な高性能プロセッサは、BPT102からの予測された分岐評価をBHR100に記憶し、予測が誤っていることが分かる場合には、誤った予測の回復オペレーションの一部分としてBHR100を後で訂正する。この図面は、明確にするために、このインプリメンテーションの特徴を反映していない。

#### 【0013】

BHR100を使用した分岐予測器の効力を低減させる可能性のある一般的なコード構造は、ループである。ループは、ループを介するたびに増分されるインデックス変数が、ループ終了値に到達しているかどうかなどのループ終了条件をテストする条件付き分岐命令と共に終了する。そうでない場合には、実行は、別の反復についてのループの始め、および別のループ終了条件付き分岐評価へと分岐して戻る。

10

#### 【0014】

ループが、多数の反復を介して実行する場合、ループ終了分岐命令の「成立」後分岐は、部分的に、または完全にBHR100を飽和させる。ループの反復の回数が、BHR100の幅に等しいか、あるいは超過する場合には、ループの端で、n - ビットBHRは、ループ反復からもたらされる長い一連の分岐が成立する評価に対応し、そしてループが終了するときに単一の不成立分岐評価と共に終了する、単一のゼロ（不成立）によって追隨される正確にn - 1個の1（成立）を含むことになる。これは、（ローカルまたはグローバルのBHR100のいずれかについての）先行する分岐評価とのすべての相関が失われるので、BHR100の効力を効果的に破壊する。この場合には、BHR100は、先行する分岐評価に対する分岐命令の相関を反映する分岐予測を含むエントリに対してではなく、（BPTインデックスロジック104に対する他の入力に応じて）与えられた分岐命令についての同じBPT102に対してマッピングする可能性が高いことになる。

20

#### 【0015】

さらに、飽和させられたBHR100は、BPT102において、エイリアシングを増大させる可能性がある。すなわち、BHR100が、BPT102に直接にインデックス付けする場合、一般的な反復を用いてループに追隨するすべての分岐命令は、同じBPT102エントリに対してマッピングすることになる。BHR100が、他の情報と組み合わせられる場合でさえ、エイリアシングの機会は、増大させられる。これは、ループに追隨する分岐命令についてだけでなく、BPT102においてそのエントリに対してエイリアスする分岐命令のすべてについて予測精度に悪影響を及ぼす。

30

#### 【0016】

ループが、BHR100の幅よりもより少ない反復を介して実行する場合、BHR100は、飽和させられず、何らかの先行する分岐評価ヒストリが、保持される。しかしながら、先行する分岐評価ヒストリを表すビットは、ループ終了分岐命令の非常に多くの「成立」結果により、BHR100の中で移動させられる。とりわけ、ループ反復の回数が、変化する場合には、これは、分岐予測に対して2つの悪影響を有する。第1に、分岐命令は、先行する分岐評価との同じ相関を取り込むためにBPT102においてずっと大きな数のエントリに対してマッピングすることになり、BHR30に影響を及ぼすループ終了分岐なしに必要とされることになるよりも大きなBPT102が、同じ数の分岐命令について同じ精度をサポートすることを必要とする。第2に、BPT102における分岐予測器は、「トレーニングする」ためにより長くかかることになり、BPT102が正確な分岐予測を提供し始める前に実行する必要のあるコードの量を増大させる。

40

#### 【0017】

本願の譲受人に譲渡され、参照によりここにおいて組み込まれている特許出願シリアル番号11/066,508号は、ループ終了分岐命令についてBHR100をアップデートすることを抑制することを提案している。これは、BHR100の飽和、または分岐予測精度に対する部分的な飽和の悪影響の一般的なものを解決する。しかしながら、それは、ループ動作と、後に続く分岐評価との間に存在する可能性のある相関を取り込み、活用することができない。

#### 【0018】

50

一般的なアプリケーションにおいて、分岐命令の評価は、先行するループの反復の回数に相関させられることができる。例えば、科学的プログラムは、ループの中のデータポイントを取り込み、ループに追従して、取り込まれたデータが解析される統計解析サブルーチンへと分岐することができる。しかしながら、ループが、少ないデータポイントを取り込むことと、小さなサンプルを与えることとを比較的少ない回数しか反復しない場合、統計解析は、信頼できない可能性があり、スキップされることができる。この場合には、統計解析サブルーチンへと分岐する条件付き命令の評価は、データ獲得ループの反復の回数に強く相関させられる。

#### 【 0 0 1 9 】

別の例においては、アプリケーションは、リスト、トランザクションログ、履歴ファイル、あるいは同様なデータ構造を介して検索するためにループ構造を使用することができる。検索パラメータにマッチングするアイテムがリストの中で頻繁に現れる場合、比較的少ないループ反復が、アイテムの所在を見出すために必要とされることになる。その結果、まれにしか現れないアイテムは、検索ループを介して多数の反復を必要とする可能性がある。後続の分岐命令の評価は、特定のアイテムがリストの中で現れる頻度に相関させられることができ、それ故にそのアイテムの所在を見出すために必要とされるループ反復の回数に相関させられる。

#### 【 0 0 2 0 】

ループ終了分岐命令評価に応じて B H R 1 0 0 のアップデートを抑制することは、ループ反復の回数と、後続の分岐命令の分岐動作との間のどのような相関も取り込むことができない。他方、ループ終了分岐命令の評価の完全な履歴を保持することは、完全にまたは部分的に B H R 1 0 0 を飽和させ、ループに先立つ分岐命令評価に対する相関をなくする。

#### 【 発明の開示 】

#### 【 概要 】

#### 【 0 0 2 1 】

分岐がループ終了分岐(loop-ending branch)であることを指し示すプロパティなど、ループに関連する条件付き分岐命令のプロパティに応じて、ループの反復の回数のカウント(a count of the number of iterations of the loop)は、保持され、ループ反復カウントを示す多ビット値(multi-bit value)が、B H R に記憶される。一実施形態においては、多ビット値は、実際のループカウントを備えることができ、この場合には、ビットの数は可変である。別の実施形態においては、ビットの数は固定され(例えば、2)、ループ反復カウントは、しきい値(thresholds)と比較することにより多ビット値にマッピングされる(mapped)。別個の反復カウントは、ネストされたループ(nested loops)のために保持されることができ、B H R に記憶される多ビット値は、内側ループだけ、外側ループだけ、あるいは両方の、ループ反復カウントを指し示すことができる。

#### 【 0 0 2 2 】

一実施形態は、分岐予測(branch prediction)の方法に関する。ループに関連する分岐命令のプロパティが、識別される(identified)。そのプロパティに応じて、その分岐命令に関連するループの反復の数を示す多ビット値が、B H R に記憶される。

#### 【 0 0 2 3 】

別の実施形態は、プロセッサに関する。本プロセッサは、条件付き分岐命令の評価を予測するように動作する分岐予測器(branch predictor)と、分岐予測器からの予測に基づいて命令を推論的にフェッチし、そして実行するように動作する命令実行パイプライン(instruction execution pipeline)とを含む。本プロセッサはまた、条件付き分岐命令の評価を記憶するように動作する B H R と、コードループの反復の回数をカウントするように動作するループカウンタと、を含む。本プロセッサは、条件付き分岐命令に関連するループの反復の回数を示す多ビット値を B H R に記憶するように動作する制御回路を、さらに含んでいる。

#### 【 詳細な説明 】

## 【 0 0 2 4 】

図 2 は、プロセッサ 1 0 の機能ブロック図を示している。プロセッサ 1 0 は、制御ロジック 1 4 に応じて、命令実行パイプライン(instruction execution pipeline) 1 2 の中で命令を実行する。一部の実施形態においては、パイプライン 1 2 は、複数のパイプラインを有するスーパースカラ(superscalar)設計とすることができる。パイプライン 1 2 は、パイプステージの形に構成された様々なレジスタまたはラッチ 1 6 と、1 つまたは複数の演算論理ユニット(Arithmetic Logic Unit) ( A L U ) 1 8 とを含む。汎用レジスタ(General Purpose Register) ( G P R ) ファイル 2 0 は、メモリ階層の最上部を備えるレジスタを提供する。

## 【 0 0 2 5 】

パイプライン 1 2 は、メモリアドレスの変換および許可が、命令側変換索引バッファ(instruction-side Translation Lookaside Buffer) ( I T L B ) 2 4 によって管理されて、命令キャッシュ(instruction cache) ( I - キャッシュ ) 2 2 から命令をフェッチする。条件付き分岐命令がパイプライン 1 2 において早期に復号化されるときに、分岐予測器 2 6 は、分岐動作を予測し、命令プリフェッチユニット(instruction prefetch unit) 2 8 に対してその予測を提供する。命令プリフェッチユニット 2 8 は、「成立」分岐予測についてのパイプライン 1 2 の中で計算される分岐ターゲットアドレスにおいて、あるいは「不成立」と予測される分岐についての次の順次アドレスにおいて I - キャッシュ 2 2 から命令を推論的にフェッチする。いずれの場合にも、プリフェッチされた命令は、推論的実行のためにパイプライン 1 2 にロードされる。

## 【 0 0 2 6 】

分岐予測器 2 6 は、分岐ヒストリレジスタ(Branch History Register) ( B H R ) 3 0 と、分岐予測器テーブル(Branch Predictor Table) ( B P T ) 3 2 と、B P T インデックスロジック 3 4 と、B H R アップデートロジック 3 6 とを含む。分岐予測器 2 6 は、1 つまたは複数の最終分岐 P C レジスタ(Last Branch PC register) 3 8 と、B H R アップデートロジック 3 6 に対して入力を供給する 1 つまたは複数のループカウンタ 3 7 とをさらに含むことができる。

## 【 0 0 2 7 】

データは、メモリアドレスの変換および許可が主要な変換索引バッファ( T L B ) 4 2 によって管理されて、データキャッシュ( D - キャッシュ ) 4 0 からアクセスされる。T L B 4 2 は、専用データ T L B とすることができ、あるいは命令とデータの両方についてアドレスの変換および許可を管理する一体化された T L B を備えることができる。さらに、プロセッサ 1 0 の様々な実施形態においては、I - キャッシュ 2 2 と D - キャッシュ 4 0 は、一体化され、あるいは統合されることができる。I - キャッシュ 2 2 および / または D - キャッシュ 4 0 における失敗は、メモリインターフェース 4 6 の制御の下でメイン(オフチップ)メモリ 4 4 に対するアクセスを引き起こす。

## 【 0 0 2 8 】

プロセッサ 1 0 は、様々な周辺デバイス 5 0 、 5 2 に対するアクセスを制御する入出力(Input/Output) ( I / O ) インターフェース 4 8 を含むことができる。当業者は、プロセッサ 1 0 の非常に多くの変形が可能であることを認識するであろう。例えば、プロセッサ 1 0 は、I キャッシュ 2 2 と D キャッシュ 4 0 のいずれかまたは両方についての第 2 のレベルの( L 2 ) キャッシュを含むことができる。さらに、プロセッサ 1 0 に示される 1 つまたは複数の機能ブロックは、特定の実施形態から省略されることができる。

## 【 0 0 2 9 】

ループ反復の多ビット表示(Multi-Bit Indications of Loop Iterations)

1 つまたは複数の実施形態によれば、分岐予測精度は、以前の分岐評価(previous branch evaluations)と以前のループの反復カウント(iteration count of previous loops)の両方に対して分岐評価を相関させることにより改善される。これは、ループ反復カウント(単数または複数)を示す多ビット値を B H R 3 0 に記憶することにより達成される。比較的少ないビットに対して潜在的に大きなループ反復カウントをマッピングすることによ

10

20

30

40

50



り、ループについての特性情報は、ループ終了分岐が、分岐予測器 26 における 1 つまたは複数の BHR30 を完全にまたは部分的に飽和させないようにしながら保存される。

#### 【0030】

このプロセスは、図 3 に流れ図として示される。命令が、フェッチされ、復号化される（ブロック 52）。命令が、条件付き分岐命令でない場合（ブロック 53）、それは、パイプラインの中で進み、プロセスは、次の命令をフェッチするために戻る（ブロック 52）。命令が、条件付き分岐命令である場合には（ブロック 53）、分岐がループ終了分岐かどうか（ブロック 54）についての決定が行われる。そうでない場合、分岐が（パイプラインの実行ステージなどにおいて）評価するときに、BHR30 は、単一ビットを使用して分岐評価を、すなわち分岐命令が「成立」として評価されるか、あるいは「不成立」として評価されるかを記録するように、アップデートされる（ブロック 56）。次いで、実行は、それぞれ分岐ターゲットアドレスまたは次の順次アドレスにおいて続行される（ブロック 64）。分岐がループ終了分岐である場合には（ブロック 54）、ループ反復カウンタが、増分される（ブロック 58）。ループが完了するときに（ブロック 60）、ループ反復の回数を示す多ビット値が、BHR30 へと書き込まれる（ブロック 62）。次いでループ反復カウンタはクリアされ（ブロック 63）、実行は、次の順次アドレスにおいて続行される（ブロック 64）。

10

#### 【0031】

両方のクエリ（ブロック 54）（すなわちループ終了分岐命令として分岐命令を識別すること）ループ反復カウンタを保持すること（ブロック 58）は、様々なやり方で達成されることができる。一実施形態においては、1 つまたは複数のループカウンタ（LC）37 は、ループ終了分岐であるように決定される条件付き分岐命令が、「成立」を評価するときに、増分される。LC37 の値、または（ここにおいて、より十分に論じられるように）そこから導き出される値は、ループ終了分岐がループの出口を指し示す「不成立」を評価するときに、BHR30 に書き込まれる。LC37 もまた、その時にクリアされる。

20

#### 【0032】

##### ループ終了分岐命令を識別すること (Identifying Loop-Ending Branch Instructions)

ループは、ループの終わりからループの始めへと後方に分岐することにより反復する。一実施形態によれば、分岐命令アドレスまたは PC よりも小さな分岐ターゲットアドレスを有するあらゆる条件付き分岐命令（すなわち、後方分岐）は、ループ終了分岐命令であるものと仮定される。ループ終了分岐を識別する方法は、簡潔性を提供する。図 4 に示されるように、分岐命令 PC は、分岐命令が実際にパイプラインの中で評価されるときにループ終了分岐検出口ジック (Loop-Ending Branch Detection logic) 35 において分岐ターゲットアドレス (BTA) と比較される。BTA < PC である場合、ループカウンタ (LC) 37 は、増分される。この実施形態は、分岐ターゲットアドレスが決定されるときにアドレス比較を必要とし、必ずしもすべての後方分岐がループ終了分岐であるとは限らないので、何らかの不正確さの影響を受ける可能性がある。

30

#### 【0033】

ループ終了分岐を検出する別の方法は、同じ分岐命令の反復される実行を認識することである。一実施形態においては、図 4 に示されるように、最終分岐 PC (Last Branch PC) (LBPC) レジスタ 38 は、実行される最終分岐命令の PC を記憶する。簡単なループの場合には、分岐命令の PC が、ループ終了分岐検出口ジック 35 における LBPC 38 とマッチングする（すなわち、分岐命令が評価される最終分岐命令であった）場合、分岐命令は、ループ終了分岐命令であるものと仮定され、LC37 が、増分される。

40

#### 【0034】

ループ内の条件付き分岐命令を含むコードは、ループ終了分岐検出を複雑にしてしまう可能性がある。例えば、アレイまたはリンクされたリストを検索するための一般的なアルゴリズムは、ループを作る。ループ内の分岐は、マッチングについて検査し、マッチングが見出される場合にループを早期に抜けるが、ループ終了分岐は、そのアレイまたはリン

50

クされたりリストの終わりについて検査する。この場合に、単一 L B P C 3 8 は、それが 2 つの分岐 P C によって交互に上書きされることになるので、ループ終了分岐を検出することができない。それに応じて、一実施形態においては、2 つ以上の L B P C レジスタ 3 8 が、(図 4 に示されるように)提供されることができ、正常に評価された分岐命令の P C が、対応する L B P C レジスタ ( L B P C <sub>0</sub>、L B P C <sub>1</sub>、... L B P C <sub>M</sub> ) 3 8 に記憶される。これにより、ループ内に他の条件付き分岐が存在する場合でさえ、ループ終了分岐の検出が可能になる。

#### 【 0 0 3 5 】

ループ終了分岐命令は、コンパイラまたはアセンブラによって静的にマーク付けされることもできる。一実施形態においては、コンパイラは、ループ終了分岐、例えば、「B R L P」のために使用されるだけの特定のタイプの分岐命令を生成する。B R L P 命令は、認識され、L C 3 7 は、B R L P 命令が実行パイプステージにおいて成立と評価するたびに、増分される。別の実施形態においては、コンパイラまたはアセンブラは、命令 o p コード内に 1 つまたは複数のあらかじめ定義されたビットを設定することなどにより、分岐命令にループ終了分岐表示(loop-ending branch indication)を埋め込むことができる。ループ終了分岐ビットは、検出され、L C 3 7 は、分岐命令が、実行パイプステージにおいて成立と評価するときに、増分される。一般に、ループ終了インジケータ(Loop-Ending Indicator)は、ループ終了分岐検出口ジック 3 5 に対して提供されることができる。ループ終了分岐の静的識別は、ループ終了識別ファンクションをコンパイラまたはアセンブラへと移動させることにより、ハードウェアと計算の複雑さを低減させる。

#### 【 0 0 3 6 】

ループ反復の回数のインジケータを生成することおよび記憶すること(Generating and Storing an Indicator of the Number of Loop Iterations)

(内側ループまたは外側ループについて)ループ終了分岐命令を識別するために使用される技法に関係なく、1 つまたは複数の実施形態によれば、ループが反復する回数の表示は、ループ反復の回数と、後続の条件付き分岐命令評価との間の任意の相関を活用するために、B H R 3 0 に記憶される。

#### 【 0 0 3 7 】

一実施形態においては、ループ反復の実際のカウンタは、B H R 3 0 に記憶される。すなわち、例えば、L C 3 7 からの多ビットカウンタは、B H R 3 0 へとシフトされ、B H R 3 0 の M S B へと先行する分岐評価を移動させる。非常に大きなループでは、このカウンタは、 $n$  個のループ反復について  $\log_2(n)$  ビットを必要とすることになり、先行する分岐評価の対応する数を移動させる。しかしながら、大きなループ反復カウンタの場合でさえも、2 進表現は、先行技術の分岐ヒストリレジスタ(すなわち、単一のゼロによって追跡される  $n - 1$  個のもの)に記憶されるループ終了分岐動作についてのかなりの圧縮である。一実施形態においては、反復カウンタは、L C 3 7 に蓄積され、B H R 3 0 へとシフトされる。別の実施形態(図示されず)においては、B H R 3 0 の少なくとも下位の  $k$  ビットが、デュアルモードシフトレジスタ/カウンタとしてインプリメントされることができ、ループ反復カウンタは、(先行する B H R 3 0 の内容が、カウンタが大きくなるにつれて左にシフトして) B H R 3 0 において直接に増分されることができる。

#### 【 0 0 3 8 】

別の実施形態においては、ループ反復の回数の固定された多ビット表示が、B H R 3 0 に記憶される。例えば、2 ビットが、以下の意味をもつように、ループ反復の回数の表示を記録するために割り付けられることができる。

#### 【 0 0 3 9 】

1 1 - ループ反復の非常に大きな回数(Very large number of loop iterations)

1 0 - ループ反復の大きな回数(Large number of loop iterations)

0 1 - ループ反復の適度な回数(Moderate number of loop iterations)

0 0 - ループ反復の小さな回数(Small number of loop iterations)

この例においては、(小さいものと適度なものとの間(0 0 - > 0 1))と、適度なもの

10

20

30

40

50

と大きなものとの間 ( 0 1 - > 1 0 ) と、大きなものと非常に大きなものとの間 ( 1 0 - > 1 1 ) の ) 3 つのしきい値は、しきい値化ロジック (Thresholding logic) 3 9 によって指し示されるように、実際のループ反復カウンタを 2 ビット表現に対してマッピングするために使用される。一実施形態においては、しきい値は、一般的なループサイズが一般に知られている場合の、ある種の埋込みアプリケーションについてなど、与えられたプロセッサインプリメンテーションについて固定されることができる。別の実施形態においては、マッピングは、しきい値が、G P R 2 0、メモリ 4 4 におけるテーブルなどから読み取られて、スケラブルとすることができる。一般的なアプリケーションにおいては、ループ反復の回数の固定された多ビット表示は、同時にループ終了分岐によって B H R 3 0 から移動させられた先行する分岐評価の回数を制限しながら、分岐評価動作との意味のある相関を可能とする、ループ反復カウンタの十分な表示を提供することができる。

10

#### 【 0 0 4 0 】

もちろん、実際のループ反復カウンタは、B H R 3 0 におけるループ反復カウンタ精度と分岐評価変位 (branch evaluation displacement) との間の望ましい、あるいは必要とされるバランスを達成するために、しきい値の対応する数を伴う任意の固定数のビット、例えば 3 ビットと 7 個のしきい値、4 ビットと 1 5 個のしきい値などに対してマッピングされることができる。

#### 【 0 0 4 1 】

##### ネストされるループ (Nested Loops)

ループは、1 つまたは複数のネストされた、あるいは内側のループを含むことができる。したがって、一実施形態においては、複数の L C カウンタ ( L C <sub>0</sub>、L C <sub>1</sub>、... L C <sub>M</sub> ) 3 7 が提供され、これらは、L B P C 3 8 レジスタの数に対応する可能性がある。L C 3 7 は、ネストされたループの反復カウンタを保持するために、成立評価と、関連する L B P C <sub>M</sub> 3 8 とのマッチングのすぐ後に増分される。

20

#### 【 0 0 4 2 】

内側ループが終了するたびに、内側ループの反復の回数の多ビット表示を記憶することにより、B H R 3 0 は、外側ループが、比較的少ない回数、反復する時まで、部分的に、あるいは完全に飽和するようになる可能性がある。したがって、一実施形態においては、ひとたびループが内側ループであると決定された後には、内側ループの初期実行の後の後続の反復カウンタは、B H R 3 0 をアップデートしないように抑制される可能性がある。この実施形態においては、外側ループが終了する時まで、B H R 3 0 は、外側ループの第 1 の反復だけについての内側ループの反復カウンタの表示と、外側ループの反復カウンタの表示とを含むことになる。

30

#### 【 0 0 4 3 】

別の実施形態においては、内側ループ反復カウンタは、外側ループ反復カウンタの表示を用いて、B H R 3 0 におけるその値を上書きすることにより、捨てられることができる。これは、後続の分岐評価の、外側ループの反復カウンタとの相関を可能としながら、最大数の先行する分岐評価を B H R 3 0 に保存する。

#### 【 0 0 4 4 】

別の実施形態においては、外側ループ反復カウンタは、外側ループが終了するときに B H R 3 0 のアップデートを抑制することにより、捨てられることができる。これは、この場合にも、後続の分岐評価の内側ループの反復カウンタとの相関を可能にしながら、最大数の先行する分岐評価を B H R 3 0 に保存し、これは、一部のアプリケーションにおいては後続の分岐評価について、より予測的なものとして行うことができる。

40

#### 【 0 0 4 5 】

条件付き分岐命令は、例えば分岐命令アドレスまたは P C と、命令タイプと、o p コードにおけるインジケータビットの存在または不在 (presence, vel non) とを含めて一般的なプロパティを有する。ここにおいて使用されるように、分岐オペレーションのプロパティ、および / または分岐に関するプログラムのプロパティは、分岐命令のプロパティと考えられる。例えば、分岐命令 P C が、1 つまたは複数の L B P C レジスタ 3 8 の内容にマ

50

ツチングするかどうかと、分岐ターゲットアドレスが、分岐命令 P C に対して前方にあるか、または後方にあるかは、分岐命令のプロパティである。

【 0 0 4 6 】

本開示は、その特定の特徴、態様、および実施形態に関して、ここに説明されてきたが、多くの変形、修正、および他の実施形態が、本開示の広い範囲内において可能であることは明らかであり、したがって、すべての変形、修正、および実施形態は、本開示の範囲内にあるものと見なされるべきである。それ故に、本実施形態は、すべての点において例示的であって限定的ではないとして解釈されるべきであり、添付の特許請求の範囲の意味する範囲および均等の範囲の中に入るすべての変更は、そこに包含されるべきであるように意図される。

【図面の簡単な説明】

【 0 0 4 7 】

【図 1】 先行技術の分岐予測器回路の機能ブロック図である。

【図 2】プロセッサの機能ブロック図である。

【図 3】分岐命令を実行する方法の流れ図である。

【図 4】 1 つまたは複数の最終分岐 PC レジスタおよびループカウンタを含む分岐予測器回路の機能ブロック図である。

10

【 図 1 】

图 1

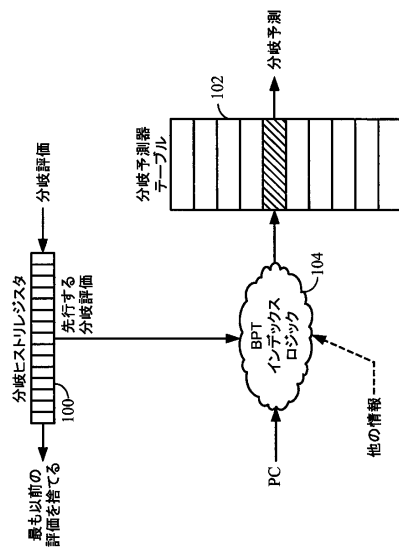


FIG. 1  
(先行技術)

【 図 2 】

图 2

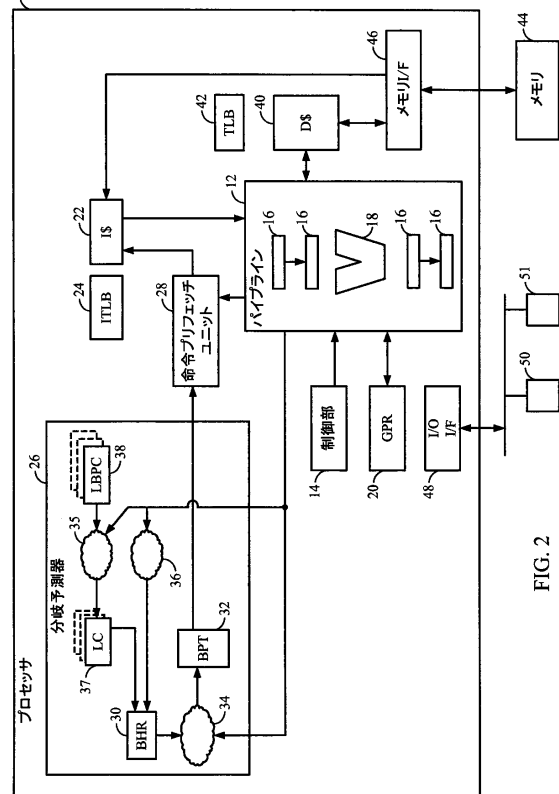
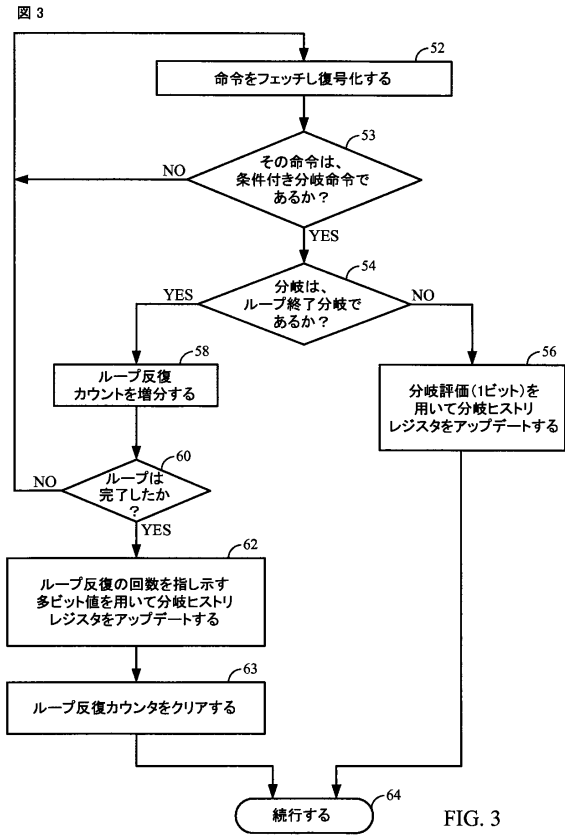
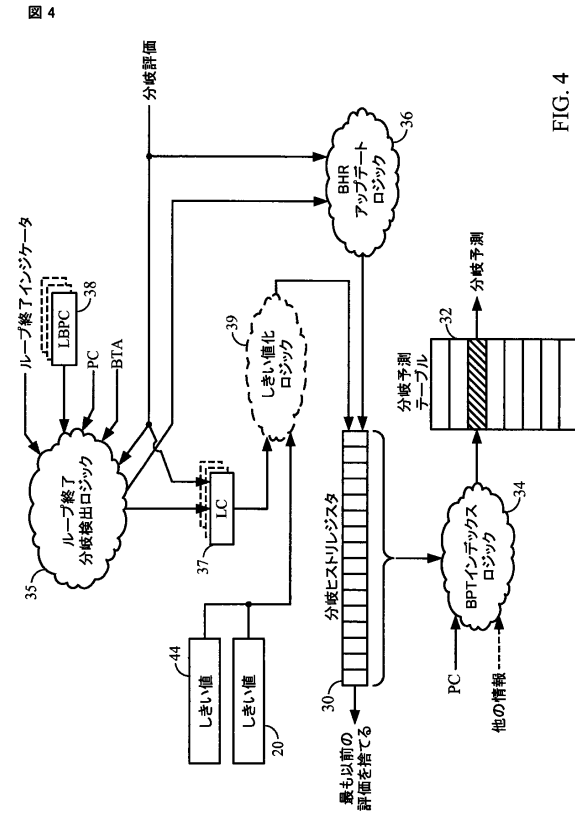


FIG. 2

【 図 3 】



【 図 4 】



## フロントページの続き

- (74)代理人 100075672  
弁理士 峰 隆司
- (74)代理人 100095441  
弁理士 白根 俊郎
- (74)代理人 100084618  
弁理士 村松 貞男
- (74)代理人 100103034  
弁理士 野河 信久
- (74)代理人 100119976  
弁理士 幸長 保次郎
- (74)代理人 100153051  
弁理士 河野 直樹
- (74)代理人 100140176  
弁理士 砂川 克
- (74)代理人 100100952  
弁理士 風間 鉄也
- (74)代理人 100101812  
弁理士 勝村 紘
- (74)代理人 100070437  
弁理士 河井 将次
- (74)代理人 100124394  
弁理士 佐藤 立志
- (74)代理人 100112807  
弁理士 岡田 貴志
- (74)代理人 100111073  
弁理士 堀内 美保子
- (74)代理人 100134290  
弁理士 竹内 将訓
- (74)代理人 100127144  
弁理士 市原 卓三
- (74)代理人 100141933  
弁理士 山下 元
- (72)発明者 ディーフェンダーファー、ジェームズ・ノリス  
アメリカ合衆国、ノースカロライナ州 27539、アベックス、インクベリー・コート 4000
- (72)発明者 リクリク、ボヒュースラブ  
アメリカ合衆国、ノースカロライナ州 27560、モリスビル、スウィート・スポット・サークル 1017

審査官 三坂 敏夫

- (56)参考文献 米国特許出願公開第2005/0283593(US, A1)  
米国特許第05909573(US, A)  
米国特許出願公開第2001/0056531(US, A1)  
特開昭63-005442(JP, A)  
特開2004-78599(JP, A)  
Po-Yung Chang et al., "Improving branch prediction accuracy by reducing pattern history table interference", Proceedings of the 1996 Conference on Parallel Architectures and

d Compilation Techniques,1996. , 米国 , IEEE , 1996年10月 , pages:48-57

中田 尚 他 , 「高性能マイクロプロセッサの高速シミュレーション」 , 先進的計算基盤システム  
シンポジウム SACSIS2003 論文集 , 日本 , 社団法人情報処理学会 , 2003年 5  
月28日 , 第2003巻 , 89頁 ~ 96頁 , 特に4.2.2章の記載を参照

(58)調査した分野(Int.Cl. , DB名)

G06F 9/38